



# Building poppin' products

the Agile way



GLENN BIEGER | PROGRAM MANAGER | ATLASSIAN

## Who am I?

## Agenda

Who am I

The birth of Agile

Building great products

Being a great team

Key takeaways

It's a great time  
to be in the  
software and  
tech industry

Who led the digital transformation of  
your company?

- A) CEO
- B) CTO
- C) COVID-19

## Software companies are disrupting entire industries



Entertainment Industry  
\$47 B Market Value



Transportation Industry  
\$40 B Market Value



IoT Industry (A Google Company)  
\$3.2 B Market Value

*Uber, the world's largest taxi company owns no vehicles, Facebook the world's most popular media owner creates no content, Alibaba, the most valuable retailer has no inventory & Airbnb the world's largest accommodation provider owns no real estate*



Travel & Hospitality Industry  
\$22 B Market Value



Social Media Industry  
\$242 B Market Value



E-commerce Industry  
\$172 B Market Value

Source: angel.co as on 08/27/2015

## Glenn Bieger



The Netherlands  
Studied Business IT & Management



12 years in tech  
Team Lead, Delivery Lead, Project & Program Manager



At Atlassian  
Program Management, Security



## Hipchat Stride

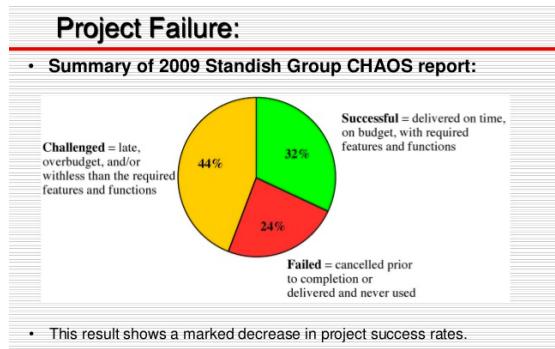
Atlassian's HipChat and Stride to be discontinued, with Slack buying up the IP

Greg Kumparak @grg / 6:33 am AEST • July 27, 2018

Comment



Even after several decades, we don't have a good record of successfully delivering software projects



Most projects or companies fail as they don't deliver the right product at the right time due to:



Not understanding the customer



Unclear or changing requirements



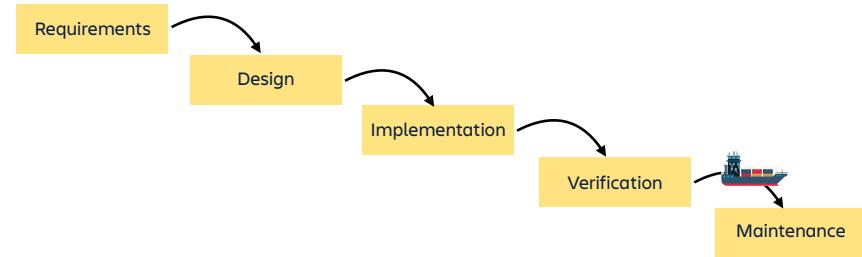
Schedule overruns



Ineffective teams

## The birth of Agile

### “Waterfall”



## Command & Control Management



**Top-down**  
Decisions are made at a (senior) management level.  
Leaders know best.



**Authoritative. Exercise power**  
People use their power to make and enforce decisions and get what they want.

## An alternative “movement”

### Mid to late 90s

Various people came up with alternative ways to build software:  
RUP, eXtreme Programming, SCRUM, Crystal

### 2001 - now

Agile has evolved significantly and is now widely adopted in the industry.

Xxx

Feb 2001

17 individuals gathered to identify the common themes, values and principles.  
Manifesto for Agile software development  
Agile Alliance

## The Agile Manifesto - Values

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

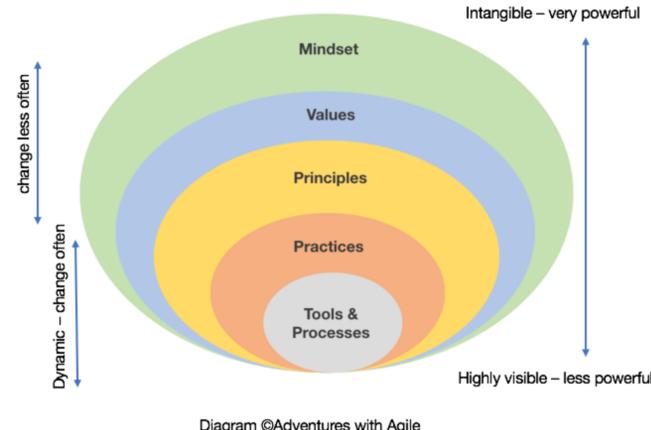
**Individuals and interactions** over processes and tools  
**Working software** over comprehensive documentation  
**Customer collaboration** over contract negotiation  
**Responding to change** over following a plan

*That is, while there is value in the items on the right, we value the items on the left more.*

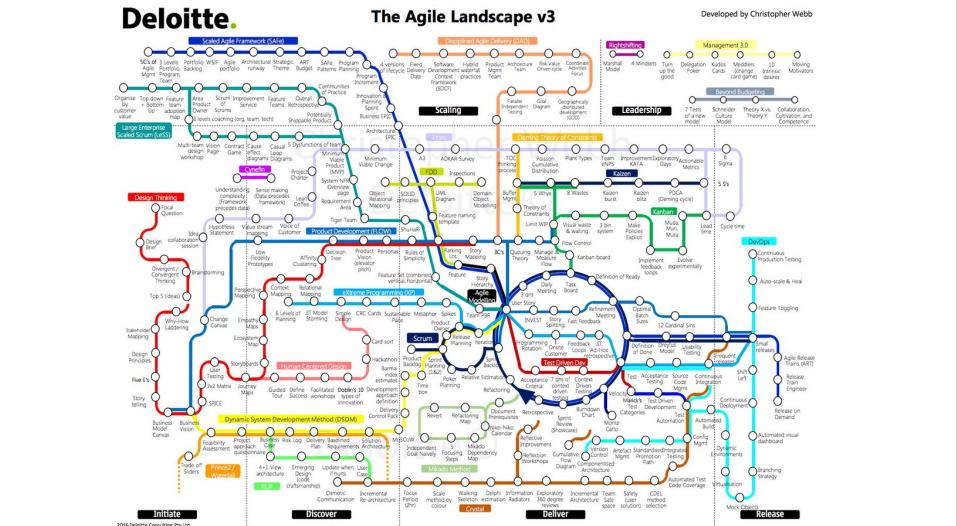
## The Agile Manifesto - Principles

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	Business people and developers must work together daily throughout the project.
Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	Working software is the primary measure of progress.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
Continuous attention to technical excellence and good design enhances agility.	Simplicity—the art of maximizing the amount of work not done—is essential.	The best architectures, requirements, and designs emerge from self-organizing teams.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

## AGILE ONION



## Deloitte.



## Incremental vs. Iterative

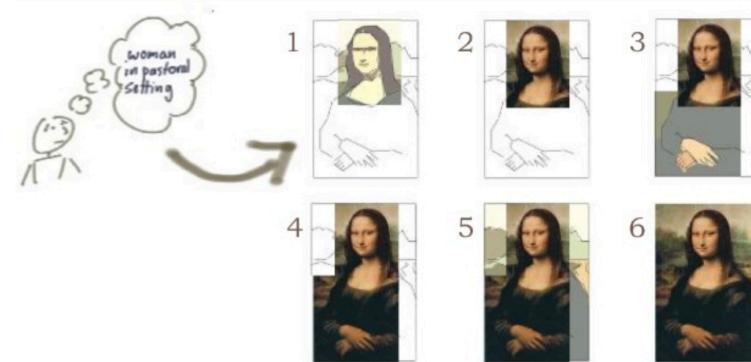
### Incremental



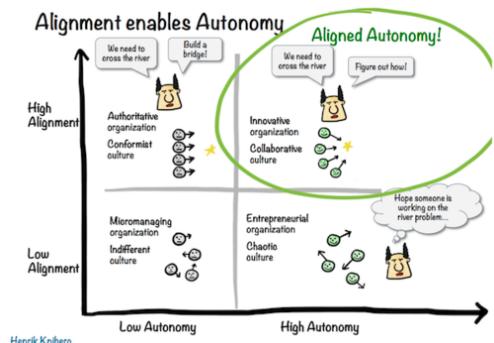
### Iterative



## Incremental and Iterative



## DIFFERENT TYPE OF LEADERSHIP



# Building great products



## GROUP ACTIVITY

### Task:

Brainstorm what makes a good product?

- Thank about an app on your phone that you like
- What make it a good app?
- Who is the target customer? What problem is it trying to solve?
- How would you improve it?

# What makes a good product?

---

## Build the right product

Focus on what the market needs first - don't get bogged down on specific features it will have or technology you will be using

## Clear value proposition

Understand your product's value proposition from the outset. How is it going to be different from what already exists in the market?

## Back it up with data

Don't just rely on intuition that your product is a good idea! Back it up with data from the market

# What makes a good product?

---

## Build the right product

Focus on what the market needs first - don't get bogged down on specific features it will have or technology you will be using

## Clear value proposition

Understand your product's value proposition from the outset. How is it going to be different from what already exists in the market?

## Back it up with data

Don't just rely on intuition that your product is a good idea! Back it up with data from the market

# What makes a good product?

---

## Build the right product

Focus on what the market needs first - don't get bogged down on specific features it will have or technology you will be using

## Clear value proposition

Understand your product's value proposition from the outset. How is it going to be different from what already exists in the market?

## Back it up with data

Don't just rely on intuition that your product is a good idea! Back it up with data from the market

# What makes a good product?

---

## Prototype, test, iterate

Get something quick that can be put in front of customers, test, measure and learn, then repeat!

## Talk to customers

Meet them, call them, ask them questions about their needs. Observe them, track their behaviour. Use customer research to help you.

## Tell a story

People engage with stories, not with facts and features - your product should tell a clear, simple, and compelling story. Founders need to be storytellers.

# What makes a good product?

---

## Prototype, test, iterate

Get something quick that can be put in front of customers, test, measure and learn, then repeat!

## Talk to customers

Meet them, call them, ask them questions about their needs. Observe them, track their behaviour. Use customer research to help you.

## Tell a story

People engage with stories, not with facts and features - your product should tell a clear, simple, and compelling story. Founders need to be storytellers.

# What makes a good product?

---

## Prototype, test, iterate

Get something quick that can be put in front of customers, test, measure and learn, then repeat!

## Talk to customers

Meet them, call them, ask them questions about their needs. Observe them, track their behaviour. Use customer research to help you.

## Tell a story

People engage with stories, not with facts and features - your product should tell a clear, simple, and compelling story. Founders need to be storytellers.

## Process for building products



### Envision it

Understand the problem and opportunity space and explore different options for how we might solve the problem.



### Make it

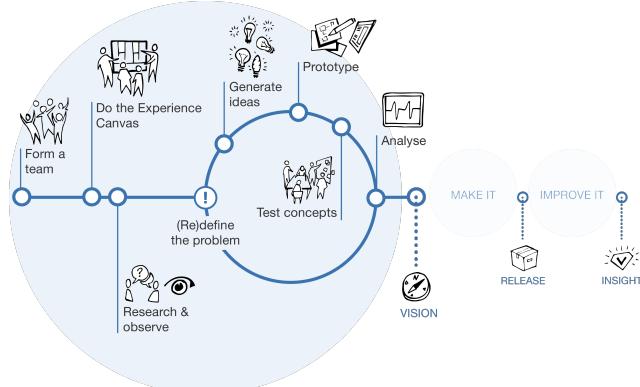
Create the first 'slice' of the solution which is production ready.



### Improve it

Measure impact and continuing to iterate on the solution.

## Envision it



Atlassian

**Experience Canvas**

Hypothesis: We think that .... will have the following effect ....

Date:  
Iteration:

Problem	Idea	Value	Stakeholders	Personas
What triggered the hypothesis?  Clearly list challenges, issues, analytics facts and assumptions	Early thoughts/options to solve this?  Best practices Patterns Comparative reviews	What is the likely user benefit and business benefit?  Expected user gains \$ business benefits Technical benefits Competitor analysis Expected analytics change	Who needs to be happy and actually has a say in it, who needs to be informed?  Driver: Team below Approver: 1 (max 3) Consulted Informed	Who will use this?  Build on the persona sets we already have or create ad-hoc personas
	<b>MVE</b>  The smallest, easiest, fastest-to-make version of your idea that you can reasonably launch as an experience.		<b>Team</b>  Keep it small and keep it balanced. Shouldn't overlap with Stakeholders above.	
<b>End to end demo</b>			<b>Test results</b>	
Tell a story end to end from the very beginning on 5 key scenarios end to end as role play, sketches, lo-fi and hi-fi prototypes, real code, the MVE			Test early and often with humans and customers Test results and recommendations	

Decision: Refine keep team / Refine split into multiple teams / Pivot, run another round / Stop

# Improving our sign up flow

A real example

## HYPOTHESIS

**By simplifying our sign up flow, we can increase the number of customers who trial our products**



## Experience Canvas

Hypothesis: We think that .... will have the following effect ....

Problem 🚧	Idea 💡	Value 💰	Stakeholders 📈	Personas 🎭
What triggered the hypothesis?  Clearly list challenges, issues, analytics facts and assumptions	Early thoughts/options to solve this?  Best practices Patterns Comparative reviews	What is the likely user benefit and business benefit?  Expected user gains \$ business benefits Technical benefits Competitor analysis Expected analytics change	Who needs to be happy and actually has a say in it, who needs to be informed?  Driver: Team below Approver: 1 (max 3) Consulted Informed	Who will use this?  Build on the persona sets we already have or create ad-hoc personas
<b>MVE 🎨</b>  The smallest, easiest, fastest-to-make version of your idea that you can reasonably launch as an experience.			<b>Team 🏁</b>  Keep it small and keep it balanced. Shouldn't overlap with Stakeholders above.	
<b>End to end demo 🎬</b>  Tell a story end to end from the very beginning on 5 key scenarios end to end as role play, sketches, lo-fi and hi-fi prototypes, real code, the MVE		<b>Test results 🎯</b>  Test early and often with humans and customers Test results and recommendations		

Decision: Refine keep team / Refine split into multiple teams / Pivot, run another round / Stop

## Data to support hypothesis

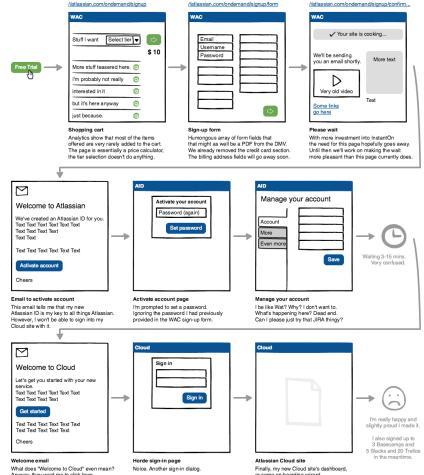
Journey	Deployment	Completion Rate (target x%)	Time on Task (current time/baseline)	Abandon Rate (meets target)
'Decide to try' to 'in product' for first time hosted customers	Cloud	NO DATA	NO DATA	NO DATA
Adding an additional product (as % of visited Applications page)	Cloud	NO DATA	NO DATA	NO DATA
Adding an add-on to an existing product (as % of visited Applications page)	Cloud	NO DATA	NO DATA	NO DATA
Cloud customer completes signup (as % of started signup)	Cloud	14.7% ⓘ	NO DATA	85.3%
Server customer completes order (as % of visited order form)	Server	9.6% ⓘ	NO DATA	90.4%
Server customer completes checkout (as % of checkout started)	Server	66.5% ⓘ	NO DATA	33.5%

Intuition - form is too complex

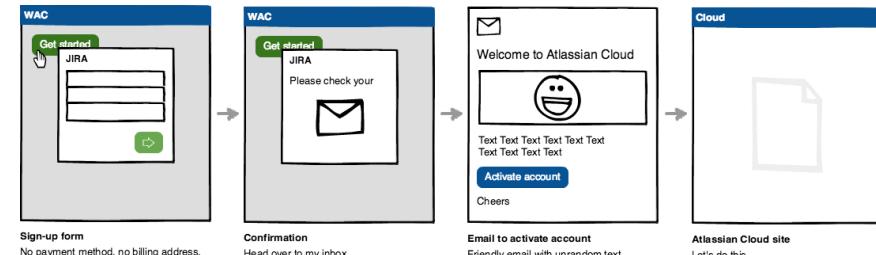
## Target persona and scenario

Will	Story	Goals
	<p>Will is asked to source an issue tracking solution for his company.  After researching the options available, Will opts to try JIRA on a colleague's recommendation.  A quick Google drop him onto WAC where he signs up for a JIRA Cloud site.  After poking around the product for a few days, Will then invites his immediate team members (3) onto the site.</p> <p><i>"If you show me the data I'll figure out this challenging problem to everyone's benefit."</i></p>	<p>Try a recommended issue tracking solutions for his company.</p>

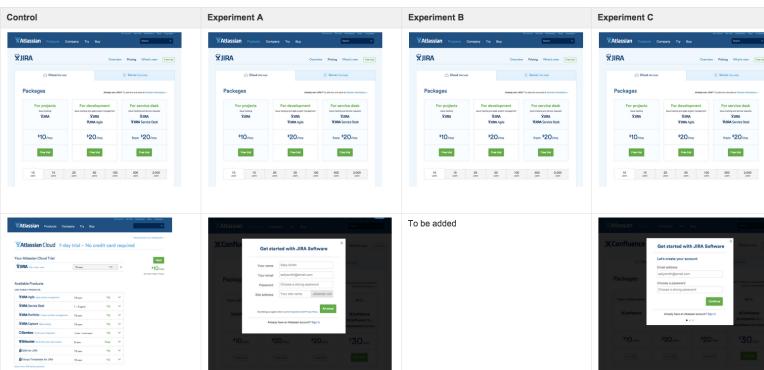
## Journey mapping shows confusing and frustrating process



We think a simpler flow will increase sign up rate



## Experiment to get early feedback



## GROUP ACTIVITY

### Envision it:

- Your team has been asked to build a cloud-only software version of a calendar app for business teams.

OR

- Your own project

Atlassian Experience Canvas	
<b>Problem</b> <span style="color: red;">*</span>	<b>Idea</b> <span style="color: green;">*</span>
What's the problem or hypothesis?	Really good idea? What's the value?
Clearly list challenges, requirements, hypotheses and assumptions	Best practices, competitive analysis, Comparative reviews
	Expected user pain
	Business benefits
	Competitor analysis
	Expected analysis design
<b>MVE</b> <span style="color: blue;">*</span>	
The most relevant, easiest, fastest to test version of your idea that you can easily pitch as an experience	
	Approve: 1-3
	Consolidate
	Refine
<b>Team</b> <span style="color: purple;">*</span>	
Keep it small and keep it simple. Don't let it overlap with Stakeholders above.	
<b>Stakeholders</b> <span style="color: orange;">*</span>	
Who needs to be happy and actually has a say in what's being built?	What's their real benefit and business benefit?
	Best practices, Competitive reviews
	Expected user pain
	Business benefits
	Competitor analysis
	Expected analysis design
<b>Personas</b> <span style="color: red;">*</span>	
Build on the persona sets we already have or create ad-hoc personas	
<b>Date</b>	
<b>Duration</b>	
<b>End to end demo</b> <span style="color: green;">*</span>	
Test a story end to end from the very beginning on	
1 key scenario end to end as use case, feature, test and UI designs	
Test early and often with humans and customers	
Test results	
Test early and often with humans and customers	
Test results and recommendations	
<b>Decision:</b> Follow this guide to make informed decisions (Not, nor another model)	

Complete an experience canvas



## Experience Canvas

Hypothesis: We think that .... will have the following effect ....

Date:  
Iteration:

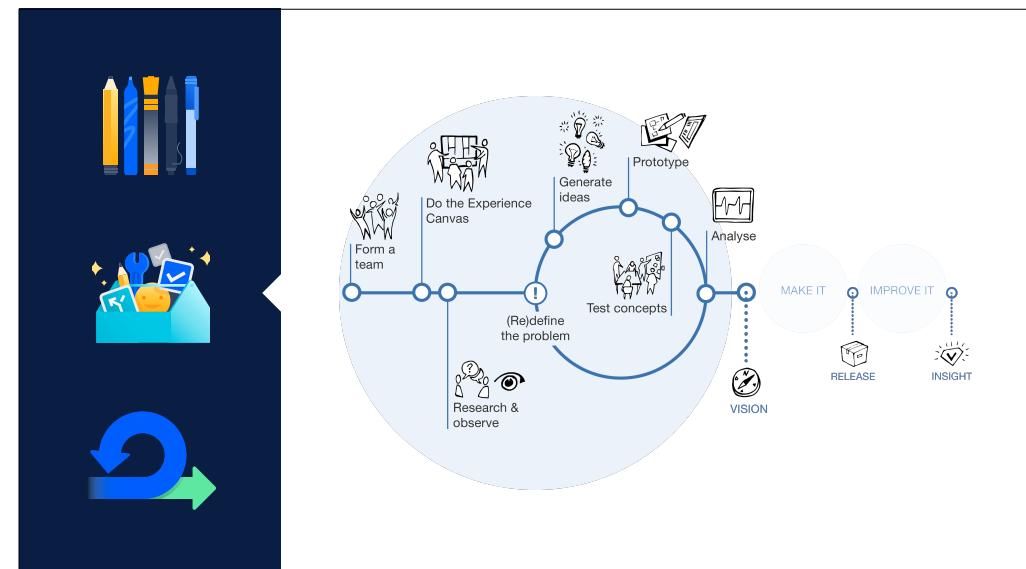
Problem 🚧	Idea 💡	Value 💰	Stakeholders 📈	Personas 🙐
What triggered the hypothesis?  Clearly list challenges, issues, analytics facts and assumptions	Early thoughts/options to solve this?  Best practices Patterns Comparative reviews	What is the likely user benefit and business benefit?  Expected user gains \$ business benefits Technical benefits Competitor analysis Expected analytics change	Who needs to be happy and actually has a say in it, who needs to be informed?  Driver: Team below Approver: 1 (max 3) Consulted Informed	Who will use this?  Build on the persona sets we already have or create ad-hoc personas
<b>MVE 🚀</b>				<b>Team 🏊</b>  Keep it small and keep it balanced. Shouldn't overlap with Stakeholders above.
<b>End to end demo 🎬</b>  Tell a story end to end from the very beginning on 5 key scenarios end to end as role play, sketches, lo-fi and hi-fi prototypes, real code, the MVE		<b>Test results 📈</b>  Test early and often with humans and customers Test results and recommendations		
<b>Decision:</b> Refine keep team / Refine split into multiple teams / Pivot, run another round / Stop				

## MINIMUM VIABLE PRODUCT

- **Based on the ideas and feedback you have got so far, describe the smallest, easiest to make version of your idea that you can launch on the app store**
- **List the incremental steps you can take to reach this launch milestone**



# Make it



# Defining requirements

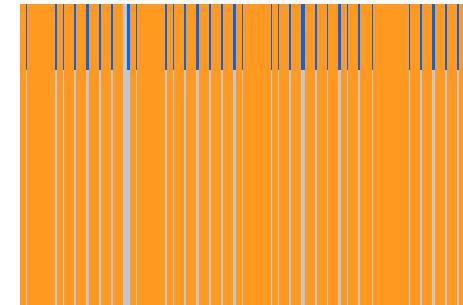
## Epics & User stories

### Epic

- Large chunk of functionality or a feature
- Lacks detail, it's not "executable"

### User story

Thin slice of working, potentially-shippable software



## User story

As a **<who>**  
I want to **<what>**  
so that **<why>**

**Who** - a user or customer or job role or persona

**What** - the goal that the user wants to accomplish or implement.

**Why** - the reason why the user needs the feature or functionality.

## User story - Examples

As a **user**,  
I want to **view all products**  
so that **I can find/buy the product I need**

As an **admin**,  
I want to **add a product**  
so that **the users can find and buy them**

As a **user**,  
I want to **search products by name**  
so that **I can more quickly find the product I need**

## Acceptance criteria

**Given**  
 <condition>  
**When**  
 <action>  
**Then**  
 <result>

**OR**

- A bullet list
- of all criteria
- that apply
- to this story

## Acceptance criteria

As a user, I want to **view all products** so that I can **find/buy the product I need**.

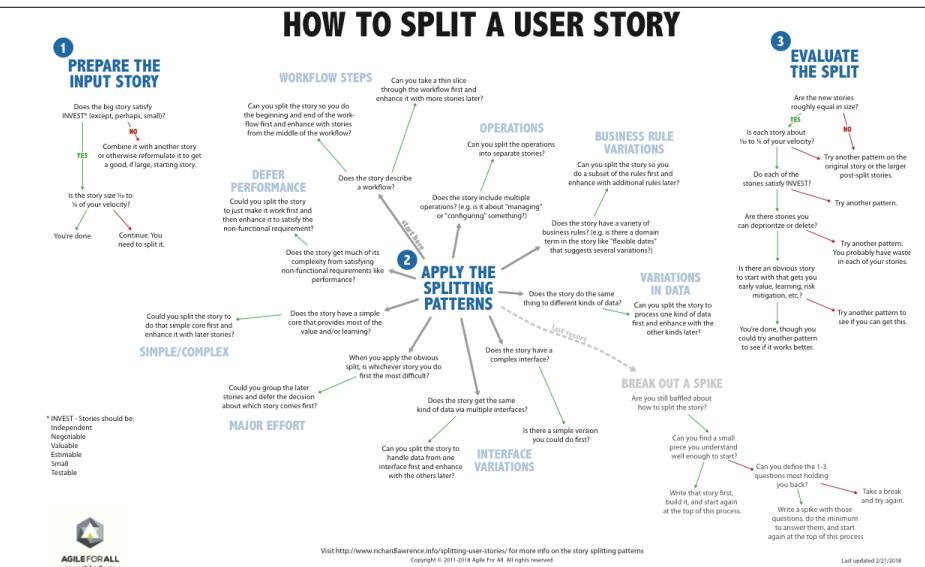
### Acceptance criteria:

**Given** I'm a logged-out user  
**When** I view the product page  
**Then** I only see "active" products

**Given** I am a returning customer  
**AND** it's boxing day  
**When** I view the product page  
**Then** the all prices will be reduced by 10%  
**AND** I see a discount Christmas banner

## Characteristics of good requirements

<b>Independent</b>	Independent of other stories (when possible), to allow for true prioritisation.
<b>Negotiable</b>	It's not a contract - it's an invitation to a conversation.
<b>Valuable</b>	It should deliver value to the user.
<b>Estimable</b>	Need to estimate/size the story to help plan & prioritise.
<b>Small</b>	<u>Small</u> slice of working software.
<b>Testable</b>	Acceptance criteria so you can objectively confirm it's done.



## GROUP ACTIVITY

Pair up with your neighbour

Pick an Epic for your/their project and work together on identifying and splitting user stories.

5 min

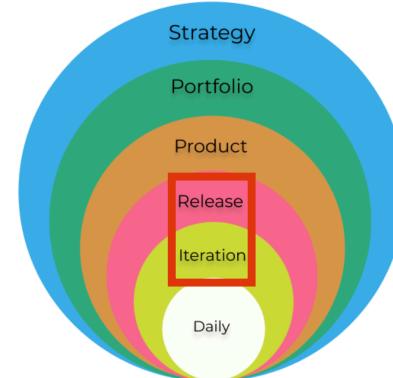
As a <who>  
I want to <what>  
so that <why>

## COMMON MYTHS

- The Product Backlog has to consist of only User Stories.
- A User Story (or a product requirement) needs to strictly follow the User Story and Acceptance Criteria format.
- Epics and User Stories is all you need - no other diagrams or documentation required.
- The product owner writes all user stories.

## Planning & estimation

### Agile Planning Onion



<https://kanbanize.com/agile/project-management/planning/>

## Different ways to estimate



Absolute estimation



Relative estimation



#NoEstimates

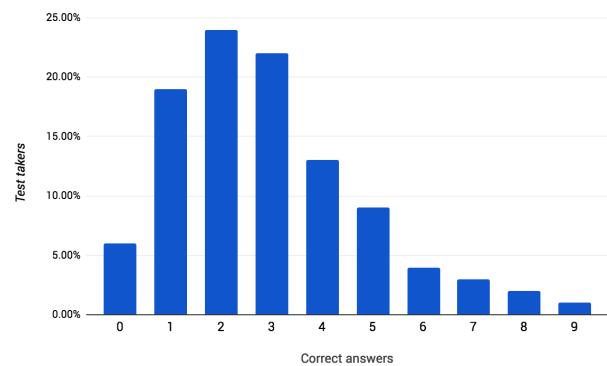
How good  
are you at  
estimating?



Working as pairs, agree on a low estimate  
and a high estimate for the following:

1. Surface temperature of the Sun
2. Latitude of Shanghai
3. Area of the Asian continent
4. The year of Alexander the Great's birth
5. Total value of US currency in circulation in 2014
6. American mean fixed internet customer data downloads per month in 2009
7. Total length of the coastline of the Pacific Ocean
8. Heaviest blue whale ever recorded
9. Number of species of birds in the world

## FALSE CONFIDENCE ABOUT ABILITY TO ACCURATELY ESTIMATE



How good  
are you at  
estimating?



Working as pairs, agree on a low estimate  
and a high estimate for the following:

1. Surface temperature of the Sun
2. Latitude of Shanghai
3. Area of the Asian continent
4. The year of Alexander the Great's birth
5. Total value of US currency in circulation in 2014
6. American mean fixed internet customer data downloads per month in 2009
7. Total length of the coastline of the Pacific Ocean
8. Heaviest blue whale ever recorded
9. Number of species of birds in the world

How good  
are you at  
estimating?



Working as pairs, agree on a low estimate  
and a high estimate for the following:

- |  |                            |
|--|----------------------------|
| 1. Surface temperature of the Sun  | 5778 K (5505 °C)           |
| 2. Latitude of Shanghai  | 31°N                       |
| 3. Area of the Asian continent   | 44,579,000 km <sup>2</sup> |
| 4. The year of Alexander the Great's birth                                   | 356 BC                     |
| 5. Total value of US currency in circulation in 2014                         | 1.29 trillion USD          |
| 6. American mean fixed internet customer data<br>downloads per month in 2009 | 48.9GB                     |
| 7. Total length of the coastline of the Pacific Ocean                        | 135,663 km                 |
| 8. Heaviest blue whale ever recorded   | 190 metric tons            |
| 9. Number of species of birds in the world                                   | 10,000                     |

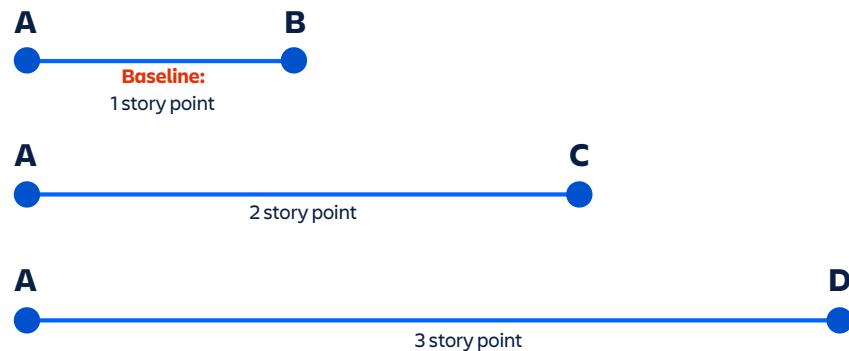
## Estimation is hard so schedules are inaccurate

### Relative estimation

#### Story points

A relative unit of measure for expressing the estimated effort required to fully implement a product backlog item or any other piece of work.

#### Story points



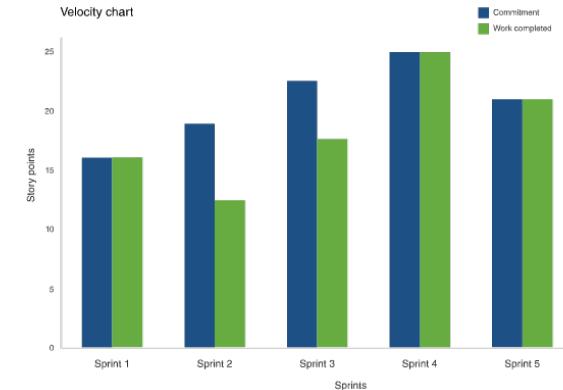
## Estimating your backlog

User story 1	3
User story 2	1
User story 3	0.5
User story 4	5
User story 5	8
User story 6	5
User story 7	2
User story 8	1
User story 9	2
User story 10	3
User story 11	13
User story 12	21

## Velocity

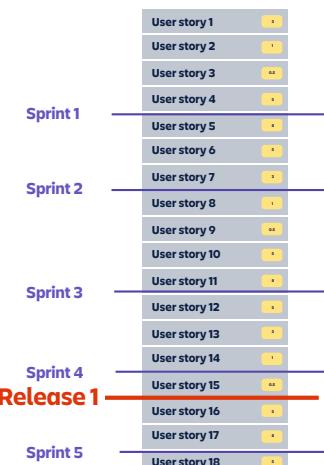
The amount work completed in a sprint

### Velocity

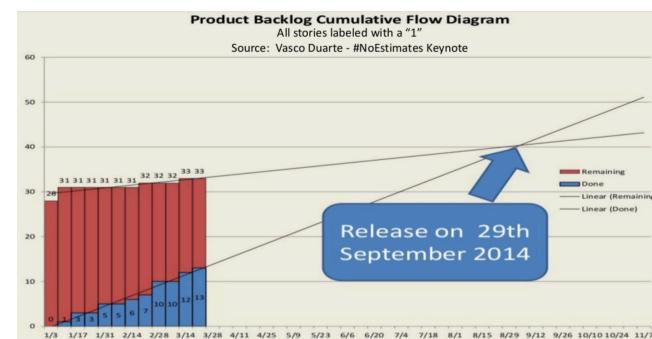


## Using velocity to plan

It will take us **5 sprints** (10 weeks) to ship Release 1



## #NoEstimates



<https://www.infoq.com/articles/noestimates-monte-carlo/>

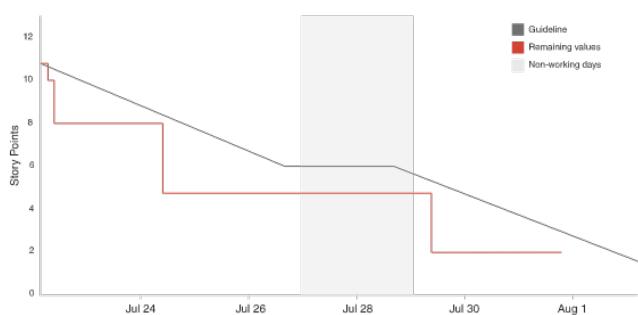
# Build it

## Scrum - one popular agile process

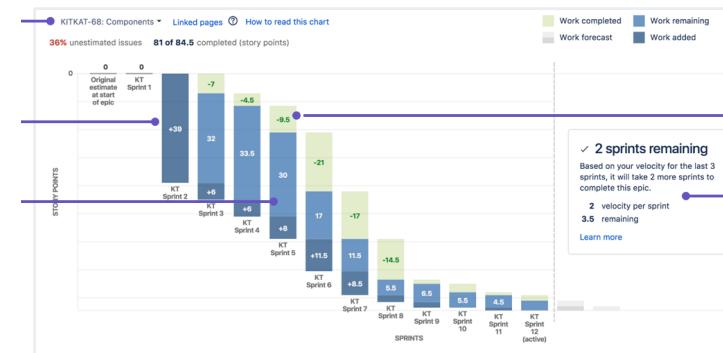


## Tracking progress during the sprint

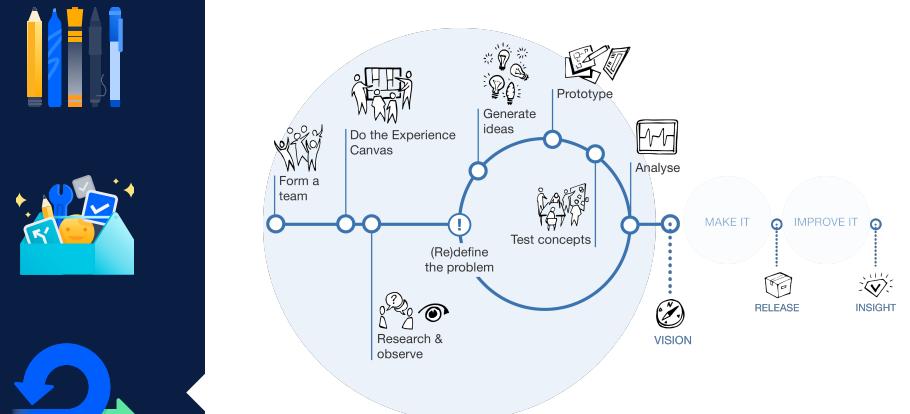
Burndown Chart



## Tracking progress during the release

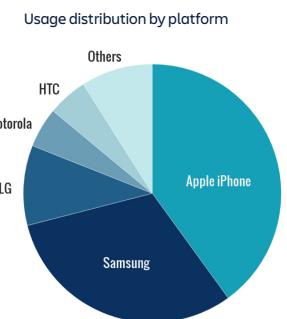
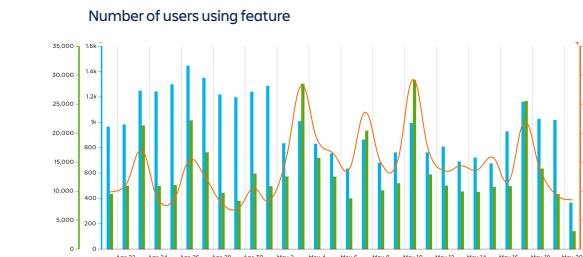


# Improve it



# The product

## MEASURE - WHAT ARE USERS ACTUALLY DOING?





Analytics tell you  
'what'.  
Customer  
research tells you  
'why'!

## The team and the process

### RETROSPECTIVES

Team is in the best place to  
improve the process

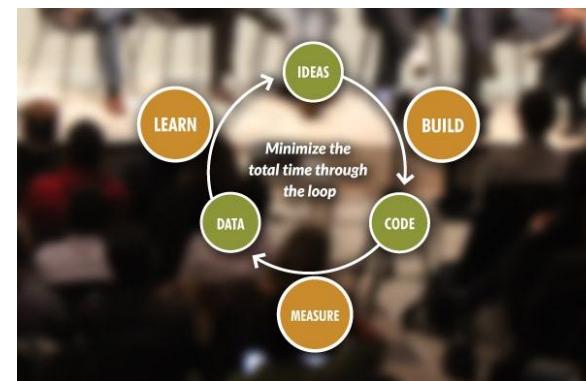
- STOP, START, CONTINUE
- 3L: Liked, Learned, Lacked
- 4L: Longed for
- Happy, Sad, Confused
- Slowed us down, make us go fast, etc.

Retro outcome:  
Define top 2 or 3 improvement areas. Clearly  
define the:

- Action
- Owner
- Due date



### KEEP REPEATING THIS PROCESS



## Being a great team

“

The major problems of our work are not so much technological as sociological in nature

**BIGGEST CHALLENGES ARE NOT USUALLY TECHNICAL**

### Setting the team up for success



Autonomous  
Cross-functional skillset



Aligned  
Know why they're here and what to do



Long-lived and stable  
Ownership, know the customer/industry and continuously improve

Trust & Collaboration

Encourage deliberate and transparent decision making

## Technical Decision Register

Create Decision

Decision	Outcome	Status	Driver	Due date
How will the Identity platform handle JIRA/Confluence user imports?	Option 1 was proposed (safe add to userbase), this should be reviewed when implementation is required (i.e. focus on server → cloud migration).	OBSOLETE		22 Jul 2016
Should we ASAP authenticate requests to Edge Authenticator?	Option 2	DECIDED	Planning Phase	05 May 2017
How do we host User Management on service-specific public URL for HCNG?	We agreed on option 3 with the proviso that the domain is not <code>atlassian.io</code> . Exact scheduling to be addressed.	DECIDED	Planning Phase	14 Oct 2016

## Make trade-offs explicit

Status	DECIDED
Outcome	Option 2
Due date	05 May 2017
Driver	
Approver	
Contributors	
Informed	

### Summary

Latency of request authentication is a major concern for Stargate + Edge Authenticator. Cryptographic operations may increase this latency. Briefly speaking any Atlassian internal consumer should be able to send valid requests to Edge Authenticator. The main concern would be excessive load on Edge Authenticator generated by unexpected consumers. Edge Authenticator Performance Tests shows that with ASAP token caching in consumers (currently just Stargate), overall latency (including p99) should not be significantly impacted.

Therefore, it is recommended that we implement ASAP authentication on Edge Authenticator endpoint, and consumers implement token caching.

### Considerations

- Latency of request authentication is a major concern for Stargate + Edge Authenticator. Cryptographic operations may increase this latency.
- Stargate → Edge Authenticator Performance Tests shows that with ASAP token caching in Stargate, overall latency (including p99) should not be significantly impacted.
- Strictly speaking any consumer should be able to send valid requests to Edge Authenticator. The main concern would be excessive load on Edge Authenticator generated by unexpected consumers.
- Ideally we only need to integrate Stargate with Edge Authenticator; however depending on timing we may need to integrate Nginx (possibly Atlassian Proxy and/or idproxy) with Edge Authenticator. Generating and caching ASA tokens in Nginx is significantly more effort than in Stargate.

### Options

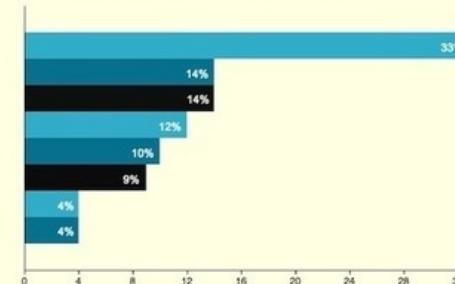
Option	Pros	Cons
1 No ASAP	<ul style="list-style-type: none"> <li>Lowest latency and CPU load on Edge Authenticator and consumers</li> </ul>	<ul style="list-style-type: none"> <li>No way to rate-limit / prevent requests from unexpected consumers.</li> </ul>
2 ASAP with consumer (Stargate) caching tokens	<ul style="list-style-type: none"> <li>Low latency and CPU load on Edge Authenticator and consumers</li> <li>Able to prevent unnecessary request processing (specifically session token validation and request ASAP token generation) from non-whitelisted consumers</li> </ul>	<ul style="list-style-type: none"> <li>Work required to cache tokens in consumers. While this should be relatively straightforward in Stargate, it is likely to be more difficult if we need to integrate Nginx with Edge Authenticator (either Atlassian Proxy or idproxy)</li> <li>Not strictly aligned with ASAP spec, but caching tokens is commonplace</li> </ul>
3 ASAP with token generated per request	<ul style="list-style-type: none"> <li>Simple on the consumer (Stargate) to implement.</li> </ul>	<ul style="list-style-type: none"> <li>Significant performance impact on consumer</li> </ul>

## Remote teams

## Communication is the greatest challenge

### WHAT'S THE GREATEST CHALLENGE?

The challenges that Agile PMs face are no longer issues of support, but are largely related to team interaction.



Poor Communication  
 Access to Expertise  
 Technical Management  
 Planning Overhead  
 Lack of Training  
 Cultural Differences  
 Team Morale  
 Lack of Support

Copyright © 2013 ProjectsAtWork All rights reserved.

## Greater reliance on tools



## Greater reliance on tools

To Do	In Progress	Done
TIS-28 Research options to travel to Pluto	TIS-27 Add Phobos and Deimos Tours as a Preferred Travel Partner	TIS-9 After 100,000 requests the SeeSpaceEZ server dies
TIS-8 Requesting available flights is now taking > 5 seconds	TIS-10 Bad JSON data coming back from hotel API	TIS-16 Establish relationship with local office supplies company
	TIS-20 Engage Saturn Shuttle Lines for group tours	TIS-7 500 Error when requesting a reservation

## Greater reliance on tools

Healthy teams



## THE TEAM HEALTH MONITOR

Assess teams across eight attributes of healthy teams

Gain a better understanding of the team's strengths

Help focus plans to address weak areas

## Key takeaways

## WHAT MAKES A HEALTHY TEAM?

### Full-time owner

There is one lead who is accountable for the result of this project. This needs to be someone whose time is at least 80% dedicated to it, and who can champion the mission inside and outside of the team.

### Balanced team

Roles and responsibilities are clear and agreed upon. The project has people with the right blend of skill set. Acknowledge that team members can change by stage.

### Shared understanding

The team has a common understanding of why they're here, the problem/need, are convinced about the idea, confident they have what they need, and trust each other.

### Value and metrics

It's clear what success means from a business and user's perspective, and there is a unique value proposition in place for the target users and to the business. Success is defined, with a goal, and how it will be measured.

### Proof of concept

Some sort of demonstration has been created and tested, that demonstrates why this problem needs to be solved, and demonstrates its value.

### One-pager

The project is summarized in a one-pager and shared with anyone so that they understand the purpose of the project, and its value.

### Managed dependencies

Clear understanding of complexity, infrastructure involved, risks, resources, effort, and timeline. Clear understanding of who we depend on, and who depends on us.

### Velocity

The team is making incremental progress by shipping concrete iterations to stakeholders (and even better to production), learning along the way, and implementing lessons learned, resulting in greater success.

## KEY TAKEAWAYS

- Building products and delivering projects is hard
- An Agile mindset and ways of working can help us
- Customer first! Always!
- Build using an iterative approach
- Culture counts - promote a healthy team

It's all about:

- Working in a **team**, as a team
- Delivering incremental **customer value**
- Focusing on **continuous improvement**

